```
#include <iostream.h>

#include <conio.h>

#include<string.h>


            char array[5][5] ;

            char plaintext[100] ;

            char ciphertext[100] ;

            void initializingArray (char[100]) ;       // fill 5*5 array

            void printArray () ;                       // print 5*5 array

            int foundChar() ;                          // found if the char I,J in array

            int foundChar(char);                       // found if the char ch in array

            char plainTextProcessing (char[100]);      // remove space and repeative char from plain text

            void getPlainText(char[100]);              // print plaintext , each two char in block

            void encryption(char[100]) ;
```



```
    void main()

            {

                for (int i=0 ; i<5 ; i++)

                    for (int j=0 ; j<5 ; j++)

                        array[i][j] = 0 ;


                    char key[100];



cout<<"\n enter the Text :\n";

cin.getline(plaintext,100,'\n');
```

```
 int n=strlen(plaintext);

cout<<"\n enter the key :\n";

cin.getline(key,100,'\n');            // take plain text from user


        initializingArray (key) ;   // fill 5*5 array

        printArray () ;                                    // print 5*5 array


        foundChar() ;

        char ch='j';                                       // found if the char I,J in array

        foundChar(ch);                             // found if the char ch in array


        plainTextProcessing (plaintext); // remove space and repeative char from plain text

        getPlainText(plaintext);                                    // print plaintext , each two char in block


        encryption(plaintext) ;

}

void getPlainText (char text[100])

{

int l=strlen(text) ;

         for (int i=0 ; i<l; i++)

         {

                if ( i % 2 == 0 &&  i != 0)             // print space between two char , and no space at begnning

                        cout << " " ;


                cout << text[i] ;

         }

        cout << "\n\n";
```

```
}


int foundChar ()

{

                for (int i=0 ; i<5 ; i++)

                        for (int j=0 ; j<5 ; j++)

                                if ( array[i][j] == 'i' || array[i][j] == 'j')

                                        return 1 ;



                return 0;

}


int foundChar (char ch)

{

                for (int i=0 ; i<5 ; i++)

                        for (int j=0 ; j<5 ; j++)

                                if ( array[i][j] == ch )

                                        return 1 ;



                return 0;

}


void initializingArray (char str[100])

{

        int i =0 ;

        int     j =0 ;

        int k = 0;
```

```
char ch = 'a' ;

int dd=strlen(str);

// fill array with key ,  didn't take repeat char


for (k=0 ; k<dd; k++)

{

        if ( j == 5 )

        {

                i++ ;

                j = 0;

        }


        if (  (str[k] == 'j'|| str[k] == 'i' ) && foundChar() )

                continue ;


        if ( foundChar(str[k]) )

                continue ;


        array[i][j++] = str[k] ;




}


// fill the remainder of array with other char , if char is repeat ignore it

for (k=0 ; k<26 ; k++ , ch++)

{

        if ( j == 5)
```

```
                  {
                          i++;
                          j = 0;
                  }

                  if (  (ch == 'j' | | ch == 'i' ) && foundChar() )
                          continue ;

                  if ( foundChar(ch)  )
                          continue ;

                  array[i][j++] = ch ;
          }
```



```
}

char plainTextProcessing(char text[100])
{
char str2;
 int len=strlen(text);
          for (int i=1;i<len;i++)
          {
                  if ( text[i] == ' ' )
                          continue ;
```

```
                if ( (text[i] == text[i-1])  &&  (i%2 != 0)  )
                        {
                        for(int ii=len;ii>i-1;ii--)
                        if(ii!=i)
                        text[ii]=text[ii-1];
                        else
                        text[ii]='x';
                        }

                str2= text[i] ;
        }

        int stlen=strlen(text);
        if ( stlen % 2 != 0 )
                text[stlen]= 'x' ;
        return str2 ;

}

void printArray ()
{
        for (int i=0 ; i<5 ; i++)
        {
                for (int j=0 ; j<5 ; j++)
                {
                        if ( array[i][j] == 'j')        { cout << array[i][j] << "\\I" << " " ; continue ; }
```

```
                        if ( array[i][j] == 'i')        { cout << array[i][j] << "\\J" << " " ; continue ; }

                        cout << array[i][j] << "   " ;

                }

                cout << "\n";

        }

}


void encryption(char text[100])

{

        int ch1Col , ch1Raw ;      // spcified [i][j] for char1

        int ch2Col , ch2Raw ;      // spcified [i][j] for char2

        char ch1 , ch2 ;

         int n=strlen(plaintext);

        for (int k=0; k<n; k+=2)

        {

                ch1 = plaintext[k] ;         // take first char in block

                ch2 = plaintext[k+1] ;     // take second char in block


                for (int i=0 ; i<5 ; i++)

                {

                        for (int j=0 ; j<5 ; j++)

                        {

                                if (array[i][j] == ch1 )

                                {

                                        ch1Col = j ;

                                        ch1Raw = i ;

                                }
```

```
                if ( array[i][j] == ch2 )
                {
                        ch2Col = j ;
                        ch2Raw = i ;
                }
        }
}

// if character in same raw
if ( ch1Raw == ch2Raw )
{
        ciphertext[k]= array[ch1Raw][(ch1Col+1)%5] ;
        ciphertext[k+1]= array[ch2Raw][(ch2Col+1)%5] ;
}

// if character in same column
else if ( ch1Col == ch2Col )
{
        ciphertext[k]= array[(ch1Raw+1)%5][ch1Col] ;
        ciphertext[k+1]= array[(ch2Raw+1)%5][ch2Col] ;
}

// if character in different raw and col
else
{
        ciphertext[k]= array[ch2Raw][ch1Col] ;
```

```
                    ciphertext[k+1]= array[ch1Raw][ch2Col] ;

            }


        }

        int st=strlen(ciphertext);

for(int gg=0;gg<st;gg++)

cout<<ciphertext[gg];

}
```